PRINCIPAL MANIFOLDS AND NONLINEAR DIMENSION REDUCTION VIA LOCAL TANGENT SPACE ALIGNMENT

ZHENYUE ZHANG* AND HONGYUAN ZHA†

Abstract. We present a new algorithm for manifold learning and nonlinear dimension reduction. Based on a set of unorganized data points sampled with noise from the manifold, the local geometry of the manifold is learned by constructing a local tangent space for each data point, and those tangent subspaces are aligned to give the internal global coordinates of the data points with respect to the underlying manifold. We also present a careful error analysis of our algorithm and show that the reconstruction errors are of second-order accuracy. We illustrate our algorithm using curves and surfaces both in 2D/3D Euclidean spaces and higher dimensional Euclidean spaces, and we also address several theoretical and algorithmic issues for further research and improvements.

Keywords: nonlinear dimension reduction, principal manifold, tangent space, subspace alignment

AMS subject classifications. 15A18, 15A23, 65F15, 65F50

1. Introduction. Many high-dimensional data in real-world applications can be modeled as data points lying close to a low-dimensional nonlinear manifold. Discovering the structure of the manifold from a set of data points sampled from the manifold possibly with noise represents a very challenging unsupervised learning problem [5]. Example low-dimensional manifolds embedded in high-dimensional input spaces include image vectors representing the same 3D objects under different camera views and lighting conditions. Another example is a set of document vectors in a text corpus dealing with a specific topic. The key observation is that the dimensions of the embedding spaces can be very high (e.g., the number of pixels for each images in the image collection or the number of terms (words and/or phrases) in the vocabulary of the text corpus), the intrinsic dimensionality of the data points, however, are rather limited due to factors such as physical constraints and linguistic correlations. Traditional dimension reduction techniques such as principal component analysis (using eigendecomposition of the sample covariance matrix) and factor analysis work usually well when the data points lie close to a *linear* (affine) subspace in the input space. They, however, tend to fail to detect nonlinear structures of the set of data points.

Recently, there have been much renewed interests in developing efficient algorithms for constructing nonlinear low-dimensional manifolds from sample data points in high-dimensional spaces, emphasizing simple algorithmic implementation and avoiding optimization problems prone to local minima [8, 11]. Two lines of research of manifold learning and nonlinear dimension reduction have emerged: one is exemplified by [11] where pairwise *geodesic* distances of the data points with respect to the underlying manifold are estimated, and the classical multi-dimensional scaling is used to project the data points into a low-dimensional space that best preserves the geodesic distances. Another line of research follows the long tradition starting with self-organizing maps (SOM) [5], principal curves/surfaces [4] and topology-preserving

* Department of Mathematics, Zhejiang University, Yuquan Campus, Hangzhou, 310027, P. R. China. zyzhang@math.zju.edu.cn. The work of this author was done while visiting Penn State University and was supported in part by the Special Funds for Major State Basic Research Projects (project G19990328), Foundation for University Key Teacher by the Ministry of Education, China, and NSF grants CCR-9901986.

† Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802, zha@cse.psu.edu. The work of this author was supported in part by NSF grants CCR-9901986.

networks [6]. The key idea is that the information about the global structure of a nonlinear manifold can be obtained from a careful analysis of the interactions of the *overlapping* local structures. In particular, the local linear embedding (LLE) method constructs a local geometric structure that is invariant to translations and orthogonal transformations in a neighborhood of each data points and seeks to project the data points into a low-dimensional space that best preserves those local geometries [8, 9].

Our approach draws inspiration from and improves upon the pioneering work in [8, 9] which opens up new directions in nonlinear manifold learning with many fundamental problems requiring to be further investigated. Our starting point is not to consider nonlinear dimension reduction in isolation as merely constructing a nonlinear projection, but rather to combine it with the process of reconstruction of the nonlinear manifold, and we argue that the two processes interact with each other in a mutually reinforcing way. we address two inter-related objectives of nonlinear structure finding: 1) to construct the so-called principal manifold [4] that goes through "the middle" of the data points; and 2) to find the global coordinate system that characterizes the set of data points in a low-dimensional space. The basic idea of our approach is to use the tangent space in the neighborhood of a data point to represent the local geometry, and then align those local tangent spaces to construct the global coordinate system for the nonlinear manifold.

The rest of the paper is organized as follows: in section 2, we formulate the problem of manifold learning and dimension reduction in more precise terms, and illustrate the intricacy of the problem using the linear case as an example. In section 3, we discuss the issue of learning local geometry using tangent spaces, and in section 4 we show how to align those local tangent spaces in order to learn the global coordinate system of the underlying manifold. Section 5 discusses how to construct the manifold once the global coordinate system is available. We call the new algorithm *local tangent space alignment* (LTSA) algorithm. In section 6, we present an error analysis of LTSA, especially illustrating the interactions among curvature information embedded in the Hessian matrices, local sampling density and noise level, and the regularity of the Jacobi matrix. In section 7, we show how the partial eigendecomposition used in global coordinate construction can be efficiently computed. We then present a collection of numerical experiments in section 8. Section 9 concludes the paper and addresses several theoretical and algorithmic issues for further research and improvements.

2. Manifold Learning and Dimension Reduction. A d-dimensional manifold $\mathcal{F}$ embedded in an m-dimensional space ($d < m$) can be represented by a function

$$f : C \subset \mathcal{R}^d \to R^m,$$

where C is compact subset of $\mathcal{R}^d$ with open interior. We are given a set of data points $x_1, \cdots, x_N$, where $x_i \in \mathcal{R}^m$ are sampled possibly with noise from the manifold, i.e.,

$$x_i = f(\tau_i) + \epsilon_i, \quad i = 1, \ldots, N,$$

where ϵ_i represents noise. By *dimension reduction* we mean the estimation of the unknown lower dimensional feature vectors τ_i's from the x_i's, i.e., the x_i's which are data points in $\mathcal{R}^m$ is (nonlinearly) projected to τ_i's which are points in $\mathcal{R}^d$, with $d < m$ we realize the objective of dimensionality reduction of the data points. By *manifold learning* we mean the reconstruction of f from the x_i's, i.e., for an arbitrary test point $\tau \in C \subset \mathcal{R}^d$, we can provide an estimate of $f(\tau)$. These two problems are inter-related, and the solution of one leads to the solution of the other. In some

situations, dimension reduction can be the means to an end by itself, and its is not necessary to learn the manifold. In this paper, however, we promote the notion that both problems are really the two sides of the same coin, and the best approach is not to consider each in isolation. Before we tackle the algorithmic details, We first want to point out that the key difficulty in manifold learning and nonlinear dimension reduction from a sample of data points is that the data points are *unorganized*, i.e., no adjacency relationship among them are known beforehand. Otherwise, the learning problem becomes the well-researched nonlinear regression problem (for a more detailed discussion, see [2] where techniques from computational geometry was used to solve error-free manifold learning problems). To ease discussion, in what follows we will call the space where the data points live the *input* space, and the space into which the data points are projected the *feature* space.

To illustrate the concepts and problems we have introduced, we consider the example of *linear* manifold learning and *linear* dimension reduction. We assume that the set of data points are sampled from a d-dimensional affine subspace, i.e.,

$$x_i = c + U\tau_i + \epsilon_i, \quad i = 1, \ldots, N,$$

where $c \in \mathcal{R}^m, \tau_i \in \mathcal{R}^d$ and $\epsilon_i \in \mathcal{R}^m$ represents noise. $U \in \mathcal{R}^{m \times d}$ is a matrix forms an orthonormal basis of the affine subspace. Let

$$X = [x_1, \cdots, x_N], \quad T = [\tau_1, \cdots, \tau_N], \quad E = [\epsilon_1, \cdots, \epsilon_N].$$

Then in matrix form, the data-generation model can be written as

$$X = c\, e^T + UT + E,$$

here e is an N-dimensional column vector of all ones. The problem of linear manifold learning is that we seek c, U and T to minimize the reconstruction error, i.e.,

$$\min \|E\| = \min_{c,U,T} \ \|X - (c\, e^T + UT)\|_F,$$

where $\|\cdot\|_F$ stands for the Frobenius norm of a matrix. This problem can be easily solved by singular value decomposition (SVD) based upon the following two observations:

1) The norm of the error matrix E can be reduced by removing the mean of the columns of E from each column of E, and hence one can assume that the optimal E has zero mean. This requirement can be fulfilled if c is chosen as the mean of X, i.e., $c = Xe/N \equiv \bar{x}$.

2) The low-rank matrix UT is the optimal rank-d approximation to the centered data matrix $X - \bar{x}e^T$. Hence the the optimal solution is given by the SVD of $X - \bar{x}e^T$,

$$X - \bar{x}e^T = Q\Sigma V^T, \quad P \in \mathcal{R}^{m \times m}, \ \ \Sigma \in \mathcal{R}^{m \times N}, \ \ V \in \mathcal{R}^{N \times N},$$

i.e., $UT = Q_d \Sigma_d V_d^T$, where $\Sigma_d = \mathrm{diag}(\sigma_1, \cdots, \sigma_d)$ with the d largest singular values of $X - \bar{x}e^T$, Q_d and V_d are the matrices of the corresponding left and right singular vectors, respectively. The optimal U^* is then given by Q_d and the learned linear manifold is represented by the linear function

$$f(\tau) = \bar{x} + U^* \tau.$$

In this model, the coordinate matrix T corresponding to the data matrix X is given by

$$T = (U^*)^T(X - \bar{x}e^T) = \operatorname{diag}(\sigma_1, \ldots, \sigma_d)V_d^T.$$

Ideally, the dimension d of the learned linear manifold should be chosen such that $\sigma_{d+1} \ll \sigma_d$.

The function f is not unique in the sense that it can be reparametrized, i.e., the coordinate can be replaced by $\tilde{\tau}$ with a global affine transformation $\tau = P\tilde{\tau}$, if we change the basis matrix U^* to U^*P. What we are interested in with respect to dimension reduction is the low-dimensional representation of the linear manifold in the feature space. Therefore, without loss of generality, we can assume that the feature vectors are uniformly distributed. For a given data set, this amounts to assuming that the coordinate matrix T is orthonormal in row, i.e., $TT^T = I_d$. Hence we we can take $T = V_d^T$ and the linear function is now the following

$$f(\tau) = \bar{x} + U^* \operatorname{diag}(\sigma_1, \ldots, \sigma_d)\tau.$$

For the linear case we just discussed, the problem of dimension reduction is solved by computing the right singular vectors V_d, and this can be done without the help of the linear function f. Similarly, the construction of the linear function f is done by computing U^* which are just the d largest left singular vectors of $X - \bar{x}e^T$.

The case for nonlinear manifolds is more complicated. In general, the global nonlinear structure will have to come from local linear analysis and alignment [8, 10]. In [8], local linear structure of the data set are extracted by representing each point x_i as a weighted linear combination of its neighbors, and the local weight vectors are preserved in the feature space in order to obtain a global coordinate system. In [10], a linear alignment strategy was proposed for aligning a general set of local linear structures. The type of local geometric information we use is the tangent space at a given point which is constructed from a neighborhood of the given point. The local tangent space provides a low-dimensional linear approximation of the local geometric structure of the nonlinear manifold. What we want to preserve are the local coordinates of the data points in the neighborhood with respect to the tangent space. Those local tangent coordinates will be aligned in the low dimensional space by different local affine transformations to obtain a global coordinate system. Our alignment method is similar in spirit to that proposed in [10]. In the next section we will discuss the local tangent space and global alignment that will be applied to data points sampled with noise in Section 4.

3. Local Tangent Space and Its Global Alignment. We assume that $\mathcal{F}$ is a d-dimensional manifold in a m-dimensional space with *unknown* generating function $f(\tau)$, $\tau \in R^d$, and we are given a data set consists of N m-dimensional vectors $X = [x_1, \ldots, x_N]$, $x_i \in \mathcal{R}^m$ and following the noise-free model,

$$x_i = f(\tau_i), \quad i = 1, \ldots, N,$$

where $\tau_i \in \mathcal{R}^d$ with $d \ll m$. The objective as we mentioned before for nonlinear dimension reduction is to reconstruct τ_i's from the corresponding function values $f(\tau_i)$'s without explicitly constructing f. Assume that the function f is smooth enough, using first-order Taylor expansion at a fixed τ, we have

$$f(\bar{\tau}) = f(\tau) + J_f(\tau) \cdot (\bar{\tau} - \tau) + O(\|\bar{\tau} - \tau\|^2), \tag{3.1}$$

where $J_f(\tau) \in \mathcal{R}^{m\times d}$ is the Jacobi matrix of f at τ. If we write the m components of $f(\tau)$ as

$$f(\tau) = \begin{bmatrix} f_1(\tau) \\ \vdots \\ f_m(\tau) \end{bmatrix}, \quad \text{then} \quad J_f(\tau) = \begin{bmatrix} \partial f_1/\partial\tau_1 & \cdots & \partial f_1/\partial\tau_d \\ \vdots & \vdots & \vdots \\ \partial f_m/\partial\tau_1 & \cdots & \partial f_m/\partial\tau_d \end{bmatrix}.$$

The tangent space $\mathcal{T}_\tau$ of f at τ is spanned by the d column vectors of $J_f(\tau)$ and is therefore of dimension at most d, i.e., $\mathcal{T}_\tau = \text{span}(J_f(\tau))$. The vector $\tau - \bar{\tau}$ gives the coordinate of $f(\tau)$ in the affine subspace $f(\tau) + \mathcal{T}_\tau$. Without knowing the function f, we can not explicitly compute the Jacobi matrix $J_f(\tau)$. However, if we know $\mathcal{T}_\tau$ in terms of Q_τ, a matrix forming an orthonormal basis of $\mathcal{T}_\tau$, we can write

$$J_f(\tau)(\bar{\tau} - \tau) = Q_\tau \theta^*_\tau,$$

Furthermore,

$$\theta^*_\tau = Q^T_\tau J_f(\tau)(\tau - \bar{\tau}) \equiv P_\tau(\bar{\tau} - \tau).$$

The mapping from τ to θ^*_τ represents a local affine transformation. This affine transformation is unknown because we do not know the function f. The vector θ^*_τ, however, has an approximation θ_τ that orthogonally projects $f(\bar{\tau}) - f(\tau)$ onto $\mathcal{T}_\tau$,

$$\theta_\tau \equiv Q^T_\tau(f(\bar{\tau}) - f(\tau)) = \theta^*_\tau + O(\|\bar{\tau} - \tau\|^2), \tag{3.2}$$

provided Q_τ is known at each τ. Ignoring the second-order term, the global coordinate τ satisfies

$$\int d\tau \int_{\Omega(\tau)} \|P_\tau(\bar{\tau} - \tau) - \theta_\tau\| d\bar{\tau} \approx 0.$$

Here $\Omega(\tau)$ defines the neighborhood of τ. Therefore, a natural way to approximate the global coordinate is to find a global coordinate τ and a local affine transformation P_τ that minimize the error function

$$\int d\tau \int_{\Omega(\tau)} \|P_\tau(\bar{\tau} - \tau) - \theta_\tau\| d\bar{\tau}. \tag{3.3}$$

This represents a *nonlinear* alignment approach for the dimension reduction problem (this idea will be picked up at the end of section 4).

On the other hand, a *linear* alignment approach can be devised as follows. If $J_f(\tau)$ is of full column rank, the matrix P_τ should be non-singular and

$$\bar{\tau} - \tau \approx P^{-1}_\tau \theta_\tau \equiv L_\tau \theta_\tau.$$

The above equation shows that the affine transformation L_τ should align this local coordinate with the *global* coordinate $\tau - \bar{\tau}$ for $f(\tau)$. Naturally we should seek to find a global coordinate τ and a local affine transformation L_τ to minimize

$$\int d\tau \int_{\Omega(\tau)} \|\bar{\tau} - \tau - L_\tau \theta_\tau\| d\bar{\tau}. \tag{3.4}$$

The above amounts to matching the local geometry in the feature space. Notice that θ_τ is defined by the "known" function value and the "unknown" orthogonal basis

matrix Q_τ of the tangent space. It turns out, however, Q_τ can be approximately determined by certain function values. We will discuss this approach in the next section. Clearly, this linear approach is more readily applicable than (3.3). We will give a detailed error analysis of the linear alignment when it is applied to sample data points with noise. Obviously, If the manifold $\mathcal{F}$ is not *regular*, i.e., the Jacobi matrix J_f is not of full column rank at some points $\tau \in C$, then the two minimization problems (3.4) and (3.3) may lead to quite different solutions.

As is discussed in the linear case, the low-dimensional feature vector τ is not uniquely determined by the manifold $\mathcal{F}$. We can reparametrize $\mathcal{F}$ using $f(g(\tau))$ where $g(\cdot)$ is a smooth 1-to-1 onto mapping of C to itself. The parameterization of $\mathcal{F}$ can be fixed by requiring that τ has a uniform distribution over C.

4. Feature Extraction through Alignment. Now we consider how to construct the global coordinates and local affine transformation when we are given a data set $X = [x_1, \ldots, x_N]$ sampled with noise from an underlying nonlinear manifold,

$$x_i = f(\tau_i) + \epsilon_i, \quad i = 1, \ldots, N,$$

where $\tau_i \in \mathcal{R}^d$, $x_i \in \mathcal{R}^m$ with $d \ll m$. For each x_i, let $X_i = [x_{i_1}, \ldots, x_{i_k}]$ be a matrix consisting of its k-nearest neighbors including x_i, say in terms of the Euclidean distance. Consider computing the best d-dimensional affine subspace approximation for the data points in X_i,

$$\min_{x,\Theta,Q} \sum_{j=1}^{k} \left\|x_{i_j} - (x + Q\theta_j)\right\|_2^2 = \min_{x,\Theta,Q} \left\|X_i - (xe^T + Q\Theta)\right\|_2^2,$$

where Q is of d columns and is orthonormal, and $\Theta = [\theta_1, \ldots, \theta_k]$. As is discussed in section 2, the optimal x is given by $\bar{x}_i$, the mean of all the x_{i_j}'s and the optimal Q is given by Q_i, the d left singular vectors of $X_i(I - ee^T/k)$ corresponding to its d largest singular values, and Θ is given by Θ_i defined as

$$\Theta_i = Q_i^T X_i (I - \frac{1}{k} ee^T) = [\theta_1^{(i)}, \cdots, \theta_k^{(i)}], \quad \theta_j^{(i)} = Q_i^T (x_{i_j} - \bar{x}_i). \tag{4.5}$$

Therefore we have

$$x_{i_j} = \bar{x}_i + Q_i \theta_j^{(i)} + \xi_j^{(i)}, \tag{4.6}$$

where $\xi_j^{(i)} = (I - Q_i Q_i^T)(x_{i_j} - \bar{x}_i)$ denotes the reconstruction error.

We now consider constructing the global coordinates $\tau_i, i = 1, \ldots, N$, in the low-dimensional feature space based on the local coordinates $\theta_j^{(i)}$ which represents the local geometry. Specifically, we want τ_{i_j} to satisfy the following set of equations, i.e., the global coordinates should respect the local geometry determined by the $\theta_j^{(i)}$,

$$\tau_{i_j} = \bar{\tau}_i + L_i \theta_j^{(i)} + \epsilon_j^{(i)}, \quad j = 1, \ldots, k, \;\; i = 1, \ldots, N, \tag{4.7}$$

where $\bar{\tau}_i$ is the mean of $\tau_{i_j}, j = 1, \ldots, k$. In matrix form,

$$T_i = \frac{1}{k} T_i ee^T + L_i \Theta_i + E_i,$$

where $T_i = [\tau_{i_1}, \ldots, \tau_{i_k}]$ and $E_i = [\epsilon_1^{(i)}, \cdots, \epsilon_k^{(i)}]$ is the local reconstruction error matrix, and we write

$$E_i = T_i(I - \frac{1}{k}ee^T) - L_i\Theta_i. \tag{4.8}$$

To preserve as much of the *local* geometry in the low-dimensional feature space, we seek to find τ_i and the local affine transformations L_i to minimize the reconstruction errors $\epsilon_j^{(i)}$, i.e.,

$$\sum_i \|E_i\|^2 \equiv \sum_i \|T_i(I - \frac{1}{k}ee^T) - L_i\Theta_i\|^2 = \min. \tag{4.9}$$

Obviously, the optimal alignment matrix L_i that minimizes the local reconstruction error $\|E_i\|_F$ for a fixed T_i, is given by

$$L_i = T_i(I - \frac{1}{k}ee^T)\Theta_i^+ = T_i\Theta_i^+, \text{ and therefore } E_i = T_i(I - \frac{1}{k}ee^T)(I - \Theta_i^+\Theta_i),$$

where Θ_i^+ is the Moor-Penrose generalized inverse of Θ_i. Let $T = [\tau_1, \ldots, \tau_N]$ and S_i be the 0-1 selection matrix such that $TS_i = T_i$,. We then need to find T to minimize the overall reconstruction error

$$\sum_i \|E_i\|_F^2 = \|TSW\|_F^2,$$

where $S = [S_1, \cdots, S_N]$, and $W = \text{ diag}(W_1, \cdots, W_N)$ with

$$W_i = (I - \frac{1}{k}ee^T)(I - \Theta_i^+\Theta_i). \tag{4.10}$$

To uniquely determine T, we will impose the constraints $TT^T = I_d$, it turns out that the vector e of all ones is an eigenvector of

$$B \equiv SWW^TS^T \tag{4.11}$$

corresponding to a zero eigenvalue, therefore, the optimal T is given by the d eigenvectors of the matrix B, corresponding to the 2nd to d+1st smallest eigenvalues of B.

Remark. We now briefly discuss the *nonlinear* alignment idea mentioned in (3.3). In particular, in a neighborhood of a data point x_i consisting of data points $X_i = [x_{i_1}, \ldots, x_{i_k}]$, by first order Taylor expansion, we have

$$X_i(I - ee^T/k) \approx J_f^{(i)}T_i(I - ee^T/k).$$

Let S_i be the neighborhood selection matrix as defined before, we seek to find $J_f^{(i)} \in \mathcal{R}m \times d$ and T to minimize

$$E(J, T) \equiv \sum_{i=1}^{N} \|(X - J_f^{(i)}T)S_i(I - ee^T/k)\|_F^2,$$

where $J = [J_f^{(1)}, \ldots, J_f^{(N)}]$. The LTSA algorithm can be considered as an approach to find an approximate solution to the above minimization problem. We can, however,

seek to find the optimal solution of $E(J,T)$ using an *alternating* least squares approach: fix J minimize with respect to T, and fix T minimize with respect to J, and so on. As an initial value to start the alternating least squares, we can use the T obtained from the LTSA algorithm. The details of the algorithm will be presented in a separate paper.

REMARK. The minimization problem (4.9) needs certain constraints (i.e., normalization conditions) to be well-posed, otherwise, one can just choose both T_i and L_i to be zero. However, there are more than one way to impose the normalization conditions. The one we have selected, i.e., $TT^T = I_d$, is just one of the possibilities. To illustrate the issue we look the following minimization problem,

$$\min_{X,Y} \|X - YA\|_F$$

The approach we have taken amounts to substitue $Y = XA^+$, and minimize $\|X(I - A^+A)\|_F$ with the normalization condition $XX^T = I$. However,

$$\|X - YA\|_F = \left\| [X,Y] \begin{bmatrix} I \\ -A \end{bmatrix} \right\|_F,$$

and we can minimize the above by imposing the normalization condition $[X,Y][X,Y]^T = I$. This nonuniqueness issue is closely related to to nonuniqueness of the parametrization of the nonlinear manifold $f(\tau)$, which can reparametrized as $f(\tau(\eta))$with a 1-to-1 mapping $\tau(\eta)$.

REMARK. Notice that the coordinates in the Θ_i's in (4.7) are obtained with respect to an orthonormal basis, therefore it seems quite natural to preserve this orthogonality in the low-dimensional feature space as well. In section A, we present some preliminary results along this line of ideas.

5. Constructing Principal Manifolds. Once the global coordinates τ_i are computed for each of the data points x_i, we can apply some non-parametric regression methods such as local polynomial regression to $\{(\tau_i, x_i)\}_{i=1}^N$ to construct the principal manifold underlying the set of points x_i. Here each of the component functions $f_j(\tau)$ can be constructed separately, for example, we have used the simple `loess` function [12] in some of our experiments for generating the principal manifolds.

In general, when the low-dimensional coordinates τ_i are available, we can construct an mapping from the τ-space (feature space) to the x-space (input space) as follows.

1. For each fixed τ, let τ_i be the nearest neighbor (i.e., $\|\tau - \tau_i\| \leq \|\tau - \tau_j\|$, for $j \neq i$). Define

$$\theta = L^{-1}(\tau - \bar{\tau}_i),$$

where $\bar{\tau}_i$ be the mean of the feature vectors in a neighbor to which τ_i belong.

2. Back in the input space, we define

$$x = \bar{x}_i + Q_i\theta.$$

Let us define by $g:\ \tau \to x$ the resulted mapping,

$$g(\tau) = \bar{x}_i + Q_i L_i^{-1}(\tau - \bar{\tau}_i). \tag{5.12}$$

To distinguish the computed coordinates τ_i from the exact ones, in the rest of this paper, we denote by τ_i^* the exact coordinate, i.e.,

$$x_i = f(\tau_i^*) + \epsilon_i^*. \tag{5.13}$$

Obviously, the errors of the reconstructed manifold represented by g depend on the sample errors ϵ_i^*, the local tangent subspace reconstruction errors $\xi_j^{(i)}$, and the alignment errors $\epsilon_j^{(i)}$. The following result show that this dependence is linear.

THEOREM 5.1. *Let* $\epsilon_i^* = x_i - f(\tau_i^*)$, $\xi_j^{(i)} = (I - Q_i Q_i^T)(x_i - \bar{x}_i)$, *and* $\epsilon_i = \tau_i - \bar{\tau}_i - L_i Q_i^T (x_i - \bar{x}_i)$. *Then*

$$\|g(\tau_i) - f(\tau_i^*)\|_2 \le \|\epsilon_i^*\|_2 + \|\xi_i\|_2 + \|L_i^{-1}\epsilon_i\|_2.$$

Proof. Substituting $L_i^{-1}(\tau_i - \bar{\tau}_i) = L_i^{-1}\epsilon_i + Q_i^T(x_i - \bar{x}_i)$ into (5.12) gives

$$\begin{aligned} g(\tau_i) &= \bar{x}_i + Q_i L_i^{-1}(\tau_i - \bar{\tau}_i) \\ &= \bar{x}_i + Q_i Q_i^T (x_i - \bar{x}_i) + Q_i L_i^{-1}\epsilon_1^{(i)}. \end{aligned}$$

Because $Q_i Q_i^T (x_i - \bar{x}_i) = x_i - \bar{x}_i - \xi_j^{(i)}$, we obtain that

$$\begin{aligned} g(\tau_i) &= x_i - \xi_j^{(i)} + Q_i L_i^{-1}\epsilon_1^{(i)} \\ &= f(\tau_i^*) + \epsilon_i^* - \xi_j^{(i)} + Q_i L_i^{-1}\epsilon_1^{(i)}. \end{aligned}$$

Therefore we have

$$\|g(\tau_i) - f(\tau_i^*)\|_2 \le \|\epsilon_i^*\|_2 + \|\xi_i\|_2 + \|L_i^{-1}\epsilon_i\|_2,$$

completing the proof. □

In the next section, we will give a detail error analysis to estimate the errors of alignment and tangent subspace approximation in terms of the noise and the geometric properties of the generating function f and the density of the generating coordinates τ_i^*. Note that ϵ_i is the first column of E_i and ξ_i the first column of $(I - Q_i Q_i^T)X_i(I - \frac{1}{k}ee^T)$.

6. Error Analysis. As is mentioned in the previous section, we assume that that the data points are generated by

$$x_i = f(\tau_i^*) + \epsilon_i^*, \quad i = 1, \ldots, N.$$

For each x_i, let $X_i = [x_{i_1}, \ldots, x_{i_k}]$ be a matrix consisting of its k-nearest neighbors including x_i in terms of the Euclidean distance. Similar to E_i defined in (4.8), we denote by E_i^* the corresponding local noise matrix, $E_i^* = [\epsilon_{i_1}^*, \ldots, \epsilon_{i_k}^*]$. The low-dimensional embedding coordinate matrix computed by the LTSA algorithm is denoted by $T = [\tau_1, \ldots, \tau_N]$. We first present a result that bounds $\|E_i\|$ in terms of $\|E_i^*\|$.

THEOREM 6.1. *Assume* $T^* = [\tau_1^*, \ldots, \tau_N^*]$ *satisfies* $(T^*)^T T^* = U_d$. *Let* $\bar{\tau}_i$ *be the mean of* $\tau_{i_1}, \cdots, \tau_{i_k}$, *Denote* $P_i = Q_i^T J_f(\bar{\tau}_i^*)$ *and* $H_{f_\ell}(\bar{\tau}_i^*)$ *the Hessian matrix of the* ℓ*-th component function of* f. *If the* P_i*'s are nonsingular, then*

$$\|E_i\|_F \le \|P_i^{-1}\|_F(\delta_i + \|E_i^*\|_F),$$

where δ_i is defined by

$$\delta_i^2 = \sum_{\ell=1}^{m}\sum_{j=1}^{k} \|H_{f_\ell}(\bar{\tau}_i^*)\|_2^2 \, \|\tau_{i_j}^* - \bar{\tau}_i^*\|_2^4$$

Furthermore, if each neighborhood is of size $O(\eta)$. Then $\|E\| \leq \|P_i^{-1}\|_F \|E^*\| + O(\eta^2)$.

Proof. First by definition (4.8), we have

$$E_i = T_i(I - \frac{1}{k}ee^T) - L_i\Theta_i = (T_i - L_iQ_i^TX_i)(I - \frac{1}{k}ee^T). \tag{6.14}$$

To represent X_i in terms of the Jacobi matrix of f, we assume that f is smooth enough and use Taylor expansion at $\bar{\tau}_i^*$, the mean of the k neighbors of τ_i^*,

$$x_{i_j} = f(\bar{\tau}_i^*) + J_i(\tau_{i_j}^* - \bar{\tau}_i^*) + \delta_j^{(i)} + \epsilon_{i_j},$$

where $J_i = J_f(\bar{\tau}_i^*)$ and $\delta_j^{(i)}$ represents the remainder term beyond the first order expansion, in particular, its ℓ-th components can be approximately written as (using second order approximation),

$$\delta_{\ell,j}^{(i)} \approx \frac{1}{2}(\tau_{i_j}^* - \bar{\tau}_i^*)^T H_{f_\ell}(\bar{\tau}_i^*)(\tau_{i_j}^* - \bar{\tau}_i^*)$$

with the Hessian matrix $H_{f_\ell}(\bar{\tau}_i^*)$ of the ℓ-th component function f_ℓ of f at $\bar{\tau}_i^*$. We have in matrix form,

$$X_i = f(\bar{\tau}_i^*)e^T + J_iT_i^*(I - \frac{1}{k}ee^T) + \Delta_i + E_i^*$$

with $\Delta_i = [\delta_1^{(i)}, \cdots, \delta_k^{(i)}]$. Multiplying by the centering matrix $I - \frac{1}{k}ee^T$ gives

$$X_i(I - \frac{1}{k}ee^T) = (J_iT_i^* + \Delta_i + E_i^*)(I - \frac{1}{k}ee^T). \tag{6.15}$$

Substituting (6.15) into (6.14) and denoting $P_i = Q_i^TJ_i$, we obtain that

$$E_i = (T_i - L_iP_iT_i^* - L_iQ_i^T(\Delta_i + E_i^*))(I - \frac{1}{k}ee^T). \tag{6.16}$$

For any $\tilde{T}$ satisfying the orthogonal condition $\tilde{T}\tilde{T}^T = I_d$ and any $\tilde{L}_i$, we also have the similar expression of (6.16) for $\tilde{T}_i$ and $\tilde{L}_i$. Note that T and L_i, $i = 1, \cdots, N$, minimize the overall reconstruction error, $\|E\|_F \leq \|\tilde{E}\|_F$. Setting $\tilde{T} = T^*$ and $\tilde{L}_i = P_i^{-1}$, we obtain the upper bound

$$\|E_i\|_F \leq \|P_i^{-1}\|_2(\|\Delta_i\|_F + \|E_i^*\|_F).$$

We estimate the norm $\|\Delta_i\|_F$ by ignoring the higher order terms, and obtain that

$$\|\Delta_i\|_F^2 \leq \sum_{\ell=1}^{m}\sum_{j=1}^{k} \|H_{f_\ell}(\bar{\tau}_i^*)\|_2^2 \, \|\tau_{i_j}^* - \bar{\tau}_i^*\|_2^4 = \delta^2,$$

completing the proof. ∎

The non-singularity of the matrix P_i requires that the Jacobi matrix J_i be of full column rank and the two subspaces span(J_i) and the d largest left singular vector space span(Q_i) are not orthogonal to each other. We now give a quantitative measurement of the non-singularity of P_i.

THEOREM 6.2. *Let $\sigma_d(\tilde{J}_i)$ be the d-th singular value of $\tilde{J}_i \equiv J_i T_i^*(I - \frac{1}{k}ee^T)$, and denote $\alpha_i = 4(\|E_i^*\|_F + \delta_i)$ with δ_i defined in Theorem 6.1. Then*

$$\|P_i^{-1}\|_F \leq (1 + \alpha_i^2)^{1/2}\|J_i\|_F.$$

Proof. The proof is simple. Let $\tilde{J}_i = U_J \Sigma_J V_J^T$ be the SVD of the matrix $\tilde{J}_i$. By (6.15) and perturbation bounds for singular subspaces [3, Theorem 8.6.5], the singular vector matrix Q_i can be expressed as

$$Q_i = (U_J + U_J^{\perp} H)(I + H_i^T H_i)^{-1/2} \tag{6.17}$$

with

$$\|H_i\|_F \leq \frac{4}{\sigma_d(\tilde{J}_i)}\Big(\|E^*\|_F + \|\Delta_i\|_F)\Big) \leq \alpha_i,$$

where $\sigma_d(\tilde{J}_i)$ is the d-largest singular value of $\tilde{J}_i$. On the other hand, from the SVD of $\tilde{J}_i$, we have $J_i T_i^* V_J = U_J \Sigma_J$, which gives

$$J_i = U_J \Sigma_J \big(T_i^* V_J\big)^{-1}.$$

It follows that

$$P_i = Q_i^T J_i = (I + H_i^T H_i)^{-1/2} \Sigma_J \big(T_i^* V_J\big)^{-1} = (I + H_i^T H_i)^{-1/2} U_J^T J_i.$$

Therefore we have

$$\|P_i^{-1}\|_F \leq (1 + \|H_i\|_F^2)^{1/2}\|J_i^+\|_F,$$

completing the proof. □

The degree of non-singularity of J_i is determined by the curvature of the manifold and the rotation of the singular subspace is mainly affected by the sample noises ϵ_j's and the neighborhood structure of x_i's. The above error bounds clearly show that reconstruction accuracy will suffer if the manifold underlying the data set has singular or near-singular points. This phenomenon will be illustrated in the numerical examples in section 8. Finally, we give an error upper for the tangent subspace approximation.

THEOREM 6.3. *Let $\mathrm{cond}(\tilde{J}_i) = \sigma_1(\tilde{J}_i)/\sigma_d(\tilde{J}_i)$ be the spectrum condition number of the d-column matrix $\tilde{J}_i$. Then*

$$\|(I - Q_i Q_i^T)X(I - \frac{1}{k}ee^T)\|_F \leq \Big(1 + 4(1 + \alpha_i^2)\mathrm{cond}(\tilde{J}_i)\Big)\,\alpha_i/4.$$

Proof. By (6.15), we write

$$(I - Q_i Q_i^T)X(I - \frac{1}{k}ee^T) = (I - Q_i Q_i^T)\tilde{J}_i + \tilde{\Delta}_i,$$

with $\|\tilde{\Delta}_i\|_F \leq \|E_i^*\|_F + \delta_i$. To estimate $\|(I - Q_i Q_i^T)\tilde{J}_i\|_F$, we use the expression (6.17) to obtain

$$\begin{aligned}(I - Q_i Q_i^T)\tilde{J}_i &= U\left(\begin{pmatrix} I \\ O \end{pmatrix} - \begin{pmatrix} I \\ H_i \end{pmatrix}(I + H_i^T H_i)^{-1}\right)\Sigma_J V_J^T \\ &= U\begin{pmatrix} H_i^T \\ -I \end{pmatrix} H_i (I + H_i^T H_i)^{-1}\Sigma_J V_J^T.\end{aligned}$$

Taking norms gives that

$$\|(I - Q_i Q_i^T)\tilde{J}_i\|_F \leq (1 + \|H_i\|_2^2)\|H_i\|_F \|\tilde{J}_i\|_2 \leq 4(1 + \alpha_i^2)(\|E_i^*\|_F + \delta_i)\mathrm{cond}(\tilde{J}_i).$$

The result required follows. □

The above results show that the accurate determination of the local tangent space is dependent on several factors: curvature information embedded in the Hessian matrices, local sampling density and noise level, and the regularity of the Jacobi matrix.

7. Numerical Computation Issues. The major computational cost of LTSA involves the computation of the smallest eigenvectors of the symmetric positive semi-defined matrix B defined in (4.11). B in general will be quite sparse because of the local nature of the construction of the neighborhoods. Algorithms for computing a subset of the eigenvectors for large and/or sparse matrices are based on computing projections of B onto a sequence of Krylov subspaces of the form

$$K_p(B, v_0) = \mathrm{span}\{v_0, Bv_0, B^2 v_0, \ldots, B^{p-1} v_0\},$$

for some initial vectors v_0 [3]. Hence the computation of matrix-vector multiplications Bx need to be done efficiently. Because of the special nature of B, Bx can be computed neighborhood by neighborhood without explicitly forming B,

$$Bx = S_1 W_1 W_1^T S_1^T x + \cdots + S_N W_N W_N^T S_N^T x,$$

where as defined in (4.10),

$$W_i = (I - \frac{1}{k})(I - \Theta_i^+ \Theta_i).$$

Each term in the above summation only involves the x_i's in one neighborhood.

The matrix $\Theta_i^+ \Theta_i$ in the right factor of W_i is the orthogonal projector onto the subspace spanned by the rows of (Θ_i). Let $\Theta_i^T = H_i R_i$ be the QR decomposition of Θ_i^T [3]. Then $\Theta_i^+ \Theta_i = H_i H_i^T$. Furthermore, we have $H^T e = 0$ because $\Theta_i e = 0$. We can rewrite W_i as

$$W_i = I - \frac{1}{k} e e^T - H_i H_i^T = I - [e/\sqrt{k},\, H_i][e/\sqrt{k},\, H_i]^T \equiv I - G_i G_i^T.$$

It is a orthogonal projector onto the null space spanned by the rows of Θ_i and $e^T/\sqrt{k}$. Therefore the matrix-vector product $y = S_1 W_1 W_1^T S_1^T x$ can be easily computed as follows: denote by $I_i = \{i_1, \cdots, i_k\}$ the set of indices for the k nearest-neighbors of x_i, then $y_j = 0$ for $j \notin I_i$ and

$$y(I_i) = T_i(I - G_i G_i^T)T_i^T x(I_i).$$

Here $y(I_i) = [y_{i_1}, \cdots, y_{i_k}]^T$ denotes the section of y determined by the neighborhood set I_i.

If one likes to compute the d smallest eigenvectors that orthogonal to e by applying some eigen-solver, the matrix B should be constructed first. The matrix B can be computed by partially (locally) summing as following

$$B(I_i, I_i) \leftarrow B(I_i, I_i) + I - G_i G_i^T, \quad i = 1, \cdots, N \tag{7.18}$$

with initial $B = 0$.

Now we are ready to present our Local Tangent Space Alignment (LTSA) algorithm.

Algorithm LTSA (Local Tangent Space Alignment). Given N m-dimensional points sampled possibly with noise from an underlying d-dimensional manifold, this algorithm produces N d-dimensional coordinates $T \in \mathcal{R}^{d \times N}$ for the manifold constructed from k local nearest neighbors.

Step 1. [Extracting local information.] For each $i = 1, \cdots, N$,
- **1.1** Determine k nearest neighbors x_{i_j} of x_i, $j = 1, \ldots, k$.
- **1.2** Compute the d left singular vector matrix Q_i of $X_i(I - \frac{1}{k}ee^T)$. Set Θ_i as in (4.5).
- **1.3** Compute the orthogonal basis matrix H_i of Θ_i^T by QR decomposition, and form $G_i = [e/\sqrt{k},\, H_i]$.

Step 2. [Constructing alingment matrix.] Form the matrix B by locally summing (7.18) if a direct eigen-solver will be used. Otherwise implement a routine that computes matrix-vector multiplication Bu for an arbitrary vector u.

Step 3. [Aligning global cordinates.] Compute the $d+1$ smallest eigenvectors of B and pick up the eigenvector matrix $[u_2, \cdots, u_{d+1}]$ corresponding to the 2nd to $d+1$st smallest eigenvalues, and set $T = [u_2, \cdots, u_{d+1}]^T$.

8. Experimental Results. In this section, we present several numerical examples to illustrate the performance of our LTSA algorithm. The test data sets include curves in 2D/3D Euclidean spaces, and surfaces in 3D Euclidean spaces. Especially, we take a closer look at the effects of singular points of a manifold and the interaction of noise levels and sample density. To show that our algorithm can also handle data points in high-dimensional spaces, we also consider curves and surfaces in Euclidean spaces with dimension equal to 100. For some of the benchmark data sets in [8], we also compare the projection results of LTSA and LLE.

First we test our LTSA method for 1D manifolds (curves) in both 2D and 3D. For a given 1D manifold $f(\tau)$ with uniformly sampled coordinates $\tau_1^*, \cdots, \tau_N^*$ in a fixed interval, we add Gaussian noise to obtain the data set $\{x_i\}$ as follows,

$$x_i = f(\tau_i^*) + \eta\, \texttt{randn}(m, 1),$$

where $m = 2, 3$ is the dimension of the input space, and `randn` is Matlab's standard normal distribution. In Figure 1, in the first row from left to right, we plot the color-

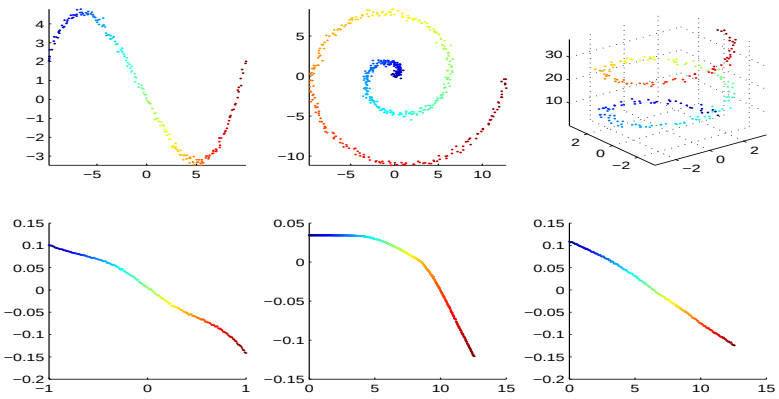

FIG. 1. *sample data points with noise from various 1-D manifolds (top) and coordinates of computed τ_i via exact τ_i^* (bottom).*

coded sample data points corresponding to the following three one-variable functions

$$\begin{aligned} f(\tau) &= (10\tau,\ 10\tau^3 + 2\tau^2 - 10\tau)^T, \quad \tau \in [-1,\ 1], \quad \eta = 0.1, \\ f(\tau) &= (\tau\cos(\tau),\ \tau\sin(\tau))^T, \qquad\quad \tau \in [0,\ 4\pi], \quad \eta = 0.2, \\ f(\tau) &= (3\cos(\tau),\ 3\sin(\tau),\ 3\tau)^T, \quad \tau \in [0,\ 4\pi], \quad \eta = 0.2. \end{aligned}$$

In the second row, we plot τ_i^* against τ_i, where τ_i's are the computed coordinates by LTSA. Ideally, the (τ_i^*, τ_i) should form a straight line with either a $\pi/4$ or $-\pi/4$ slope.

As we have shown in the error analysis in section 6, it will be difficult to align the locale tangent information Θ_i if some of the P_i's defined in (3.2) are close to be singular. One effect of this is that the computed coordinates τ_i and its neighbors may be compressed together. To clearly demonstrate this phenomenon, we consider the following function,

$$f(\tau) = [\cos^3(\tau),\ \sin^3(\tau)]^T, \quad \tau \in [0,\ \pi].$$

The Jacobi matrix (now a single vector since $d = 1$) given by

$$J_f(\tau) = 1.5\sin(2\tau)[-\cos(\tau),\ \sin(\tau)]^T$$

is equal to zero at $\tau = \pi/2$. In that case the θ-vector Θ_i defined in (4.5) will be computed poorly in the presence of noise. Usually the corresponding Θ_i will be small which also results in small τ_i and the neighbors of τ_i will also be small. In the first column of Fig 2, we plot the computed results for this 1-D curve. We see clearly near the singular point $\tau = \pi/2$ the computed τ_i's become very small, all compressed to a small interval around zero. In the second column of Fig 2, we examine another 1D curve defined by

$$f(\tau) = [10\cos(\tau),\ \sin(\tau)]^T, \quad \tau \in [\pi/2,\ 3\pi/2].$$

We notice that similar phenomenon also occurs near the point $\tau = \pi$ where the *curvature* of the curve is large, the computed τ_i's near the corresponding also become very small, clustering around zero.

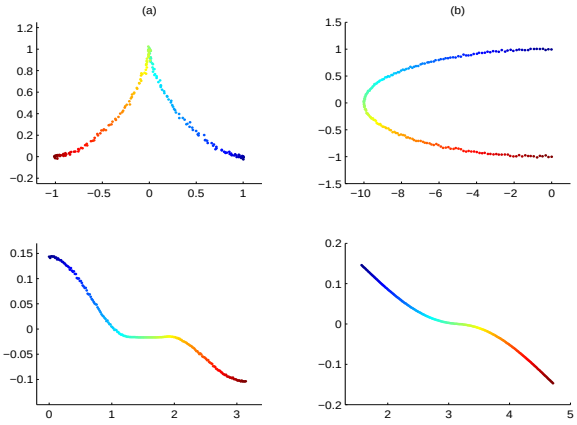

FIG. 2. *1-D manifolds with singular points (left) and corresponding coordinates τ_i via exact τ_i^* (right).*

Next we look at the issues of the interaction of sampling density and noise levels. If there are large noises around $f(\tau_i)$ relative to the sampling density near $f(\tau_i)$, the resulting centered local data matrix $X_i(I - \frac{1}{k}ee^T)$ will not be able to provide a good local tangent space, i.e., $X_i(I - \frac{1}{k}ee^T)$ will have singular values σ_d and σ_{d+1} that are close to each other. This will result in a nearly singular matrix $P_i = Q_i^T J_i$, and when plotting τ_i^* against τ_i, we will see the phenomenon of the computed coordinates τ_i getting compressed, similar to the case when the generating function $f(\tau)$ has singular and/or near-singular points. However, in this case, the result can usually be improved by increasing the number of neighbors used for producing the shifted matrix $X_i(I - \frac{1}{k}ee^T)$. In Fig 3, we plot the computed results for the generating function

$$f(\tau) = 3\tau^3 + 2\tau^2 - 2\tau, \quad \tau \in [-1.1, 1].$$

The data set is generated by adding noise in a relative fashion,

$$x_i = f(\tau_i)(1 + \eta\epsilon_i)$$

with normally distributed ϵ_i. The first three columns in Fig 3 correspond to the noise levels $\eta = 0.01$, $\eta = 0.03$, and $\eta = 0.05$, respectively. For the three data sets, We use the same number of neighbors, $k = 10$. With the increasing noise level η, the computed τ_i's get expressed at points with relatively large noise. The quality of the computed τ_i's can be improved if we increase the number of neighbors as is shown on the column (d) in Fig 3. The improved result for the same data set in column (c) with $k = 20$ used.

As we have shown in Fig 3 (column (d)), different neighborhood size k will produce different embedding results. In general, k should be chosen to match the sampling density, noise level and the curvature at each data points so as to extract an accurate local tangent space. Too few neighbors used may result a rank-deficient tangent space and leads to over-fitting, while too large a neighborhood will introduce too much bias and the computed tangent space will not match the local geometry well. It is therefore worthy of considering variable number of neighbors that are adaptively chosen at each data point. Fortunately, our LTSA algorithm seems to be less sensitive to the choice of k than LLE does as will be shown later.

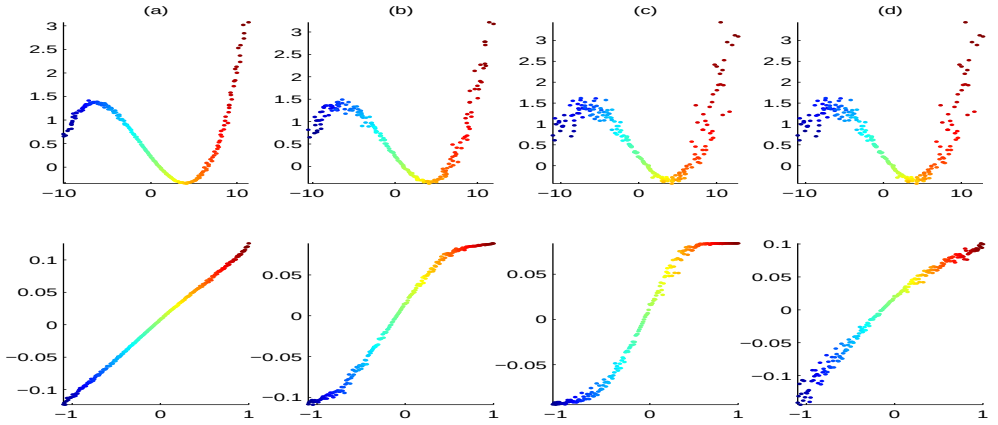

FIG. 3. *1-D manifolds with different noise levels (top) and computed coordinates τ_i vs. exact τ_i^* (bottom).*

We applied both LTSA and LLE to the S-curve data set [8] (total data points = 2000 uniformly sampled without noise) with different number of neighbors. For $d = 2$, and k which is chosen from $k = 6$ to $k = 30$, LTSA always produces coordinates T that has similar geometric structure as the generating coordinates. There are little geometric deformations in the coordinates generated by LTSA, see Figure 5. In Figure 4, we plot the results for LLE, the deformations (stretching and compression) in the generated coordinates are quite prominent. Similar results are plotted for the swissroll data set [11] in Figure 6 (LLE) and Figure 7 (LTSA). Both of these two surfaces have zero Gaussian curvature, and therefore they can be flattened without any geometric deformation, i.e., the two surfaces are *isometric* to a 2D plane.

We now apply LTSA to a 2-D manifold embedded in a 100 dimensional space. The data points are generated as follows. First we generate $N = 5000$ 3D points,

$$x_i = (t_i,\ s_i,\ h(t_i, s_i))^T + 0.01\eta_i$$

with t_i and s_i uniformly distributed in the interval $[-1,\ 1]$, the η_i's are standard normal. The $h(t, s)$ is a peak function defined by

$$h(t,s) = 0.3(1-t)^2 e^{-t^2-(s+1)^2} - (0.2t - t^3 - s^5)e^{-t^2-s^2} - 0.1e^{-(t+1)^2-s^2}.$$

This function is plotted in the left of Figure 8. We generate two kinds of data points x_i^Q and x_i^H in 100D space,

$$x_i^Q = Qx_i, \quad x_i^H = Hx_i,$$

where Q is a random orthogonal matrix resulting in an orthogonal transformation and H a matrix with its singular values uniformly distributed in $(0,\ 1)$ resulting in an affine transformation. Figure 8 plots the coordinates for x_i^Q (middle) and x_i^H (right).

One advantage of LTSA over LLE is that using LTSA we can potentially detect the intrinsic dimension of the underlying manifold by analyzing the local tangent space structure. In particular, we can examine the distribution of singular values of the data matrix X_i consisting of the data points in the neighborhood of each data point x_i. If the manifold is of dimension d, then X_i will be close to a rank-d matrix.

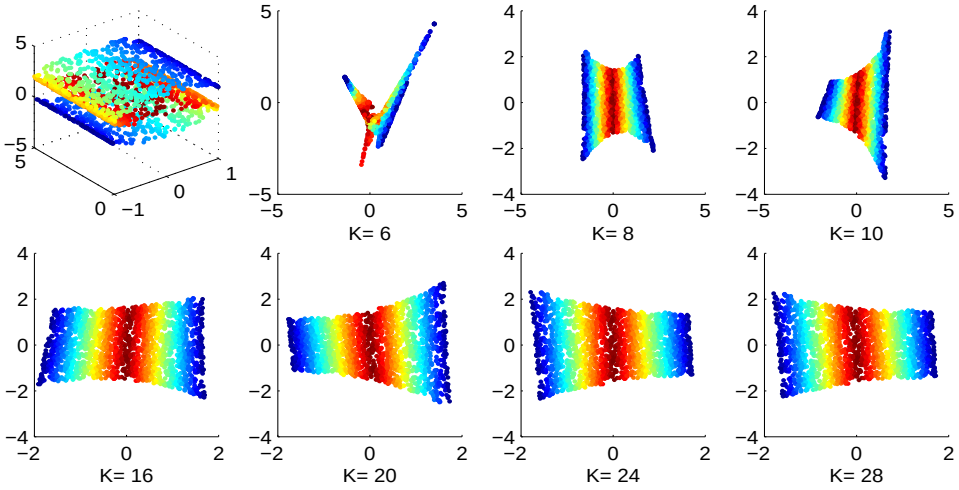

FIG. 4. *Computed 2D coordinates of the S-curve by LLE with various neighborhood size k.*

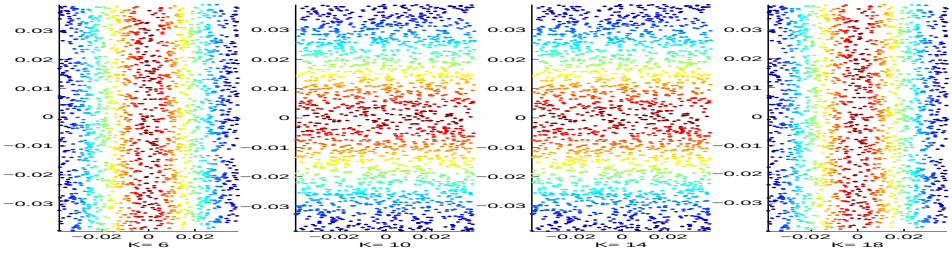

FIG. 5. *Computed coordinates of the S-curve by LTSA with various neighborhood size k.*

We illustrate this point below. The data points are x_i^Q of the 2D peak manifold in the 100D space. For each local data matrix X_i, let $\sigma_{j,i}$ be the j-the singular value of the centered matrix $X_i(I - \frac{1}{k}ee^T)$. Define the ratios

$$\rho_i^{(j)} = \frac{\sigma_{j+1,i}}{\sigma_{j,i}}.$$

In Fig 9, we plot the rations $\rho_i^{(1)}$ and $\rho_i^{(2)}$. It clearly shows the feature space should be 2-dimensional.

Next, we discuss the issue of how to use the global coordinates τ_i's as a means for clustering the data points x_i's. The situation is illustrated by Figure 10. The data set consists of three bivariate Gaussians with covariance matrices $0.2I_2$ and mean vectors located at $[1,1], [1,-1], [-1,0]$. There are 100 sample points from each Gaussian. The thick curve on the right panel represents the principal curve computed by LTSA and the thin curve by LLE. It is seen that the thick curve goes through each of the Gaussians in turn, and the corresponding global coordinates (plotted in the middle panel) clearly separate the three Gaussians. LLE did not perform as well, mixing two of the Gaussians.

The selection of the set of points to estimate the local tangent space is very crucial to the success of the algorithm. Ideally, we want this set of points to be close to the tangent space. However, with noise and/or at the points where the curvature

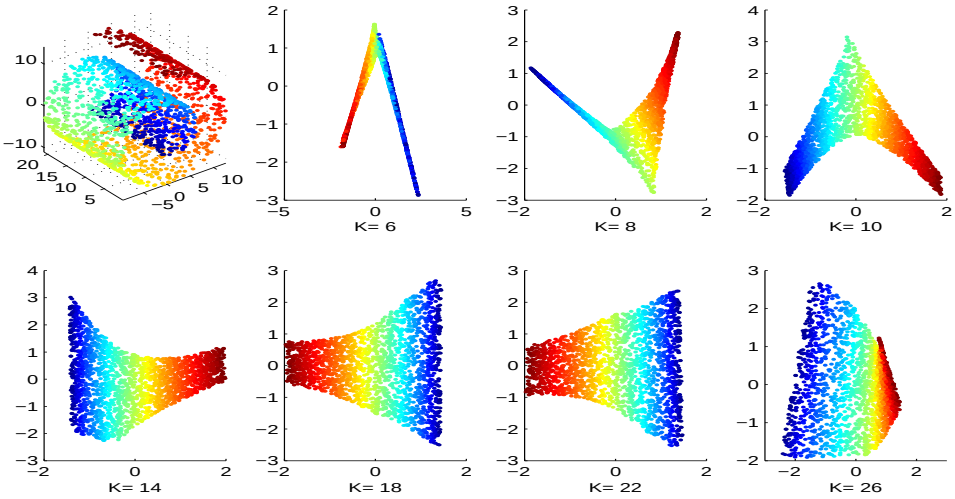

FIG. 6. *Computed 2D coordinates of the swissroll by LLE with various neighborhood size k.*

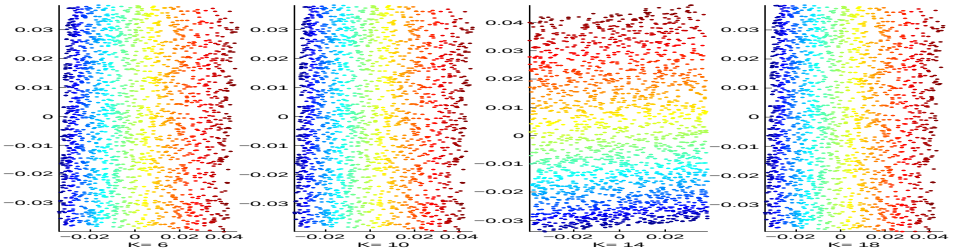

FIG. 7. *Computed coordinates of the swissroll by LTSA with various neighborhood size k.*

of the manifold is large, this is not an easy task. One line of ideas is to do some preprocessing of the data points to construct some *restricted* local neighborhoods. For example, one can first compute the minimum Euclidean spanning tree for the data set, and restrict the neighbors of each point to those that are linked by the branches of the spanning tree. This idea has been applied in self-organizing map [5]. Another idea is to use iterative-refinement, combining the computed τ_i's with the x_i's for neighborhood construction in another round of nonlinear projection. The rationale is that τ_i's as the computed global coordinates of the nonlinear manifold may give a better measure of the local geometry. An example using iterative-refinement is shown in Figure 11, the data points are sampled from the left half of a very flat ellipse (the long axis is the x-axis), one iterative-refinement gave a much better result.

Last we look at the results of applying LTSA algorithm to the face image data set [11]. The data set consists of a sequence 698 64-by-64 pixel images of a face rendered under various pose and lighting conditions. Each image is converted to an $m = 4096$ dimensional image vector. We apply LTSA with $k = 12$ neighbors and $d = 2$. The constructed coordinates are plotted in the middle of Figure 12. We also extracted four paths along the boundaries of of the set of the 2D coordinates, and display the corresponding images along each path. It can be seen that the computed 2D coordinates do capture very well the pose and lighting variations in a continuous way.

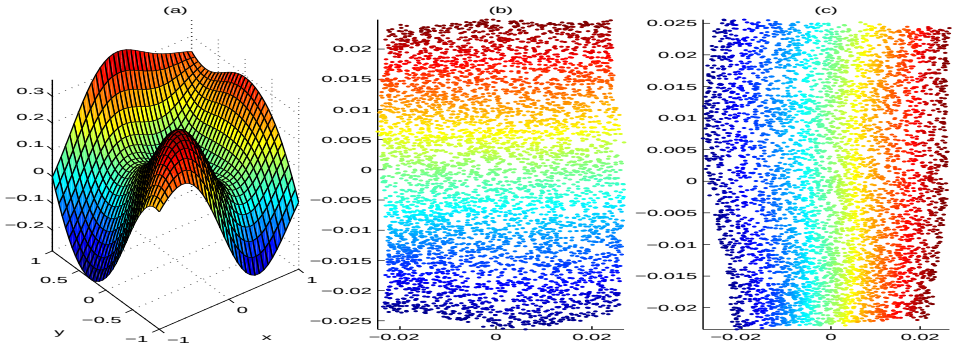

FIG. 8. *2-D manifold in a 100-D space generated by 3-D peak function: (a) 3-D peak curve, (b) coordinates for orthogonal transformed manifold, (c) coordinates for affine transformed manifold.*

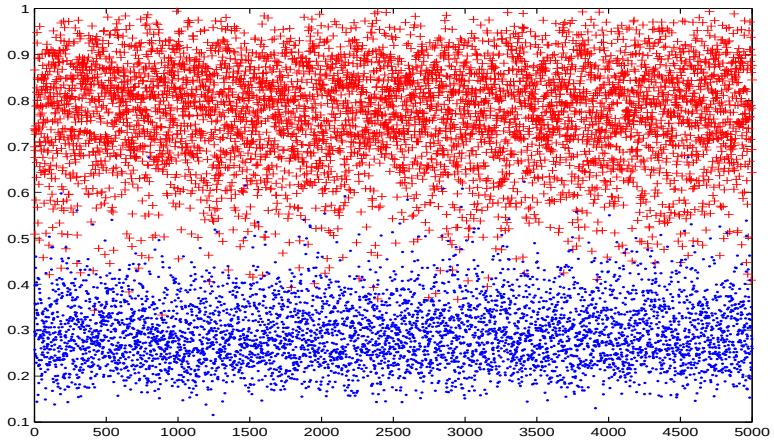

FIG. 9. *Singular value ratios* $\rho_i^{(1)}$ *(+-dots) and* $\rho_i^{(2)}$ *(·-dots).*

9. Conclusions and Feature Works. In this paper, we proposed a new algorithm (LTSA) for nonlinear manifold learning and nonlinear dimension reduction. The key techniques we used are construction of local tangent spaces to represent local geometry and the global alignment of the local tangent spaces to obtain the global coordinate system for the underlying manifold. We provide some careful error analysis to exhibit the interplay of approximation accuracy, sampling density, noise level and curvature structure of the manifold. In the following, we list several issues that deserve further investigation.

1. To make LTSA (similarly LLE) more robust against noise, we need to resolve the issue where several of the smallest eigenvalues of Θ are about the same magnitude. This can be clearly seen when the manifold itself consists of several disjoint components. If this is the case, one needs to break the matrix Θ into several diagonal blocks, and apply LTSA to each of the block. However, with noise, the situation becomes far more complicated, several eigenvectors corresponding to near-zero eigenvalues can mix together, the information of the global coordinates seems to be contained in the eigen-subspace, but how to unscramble the eigenvectors to extract the global coordinate information needs more careful analysis of the eigenvector matrix of Θ and various models of the noise. Some preliminary results on this problem have been

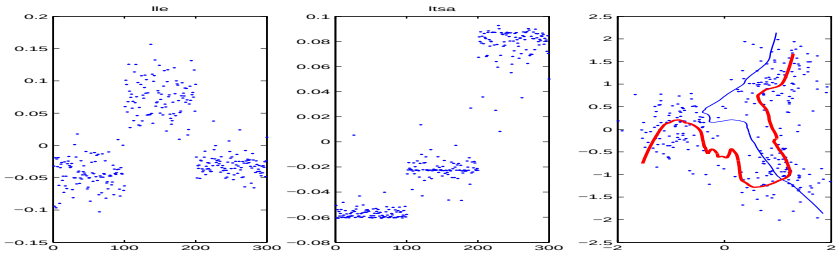

FIG. 10. *(left) Global coordinates by LLE, (middle) global coordinates by LTSA, (right) Three Gaussian data with principal curves*

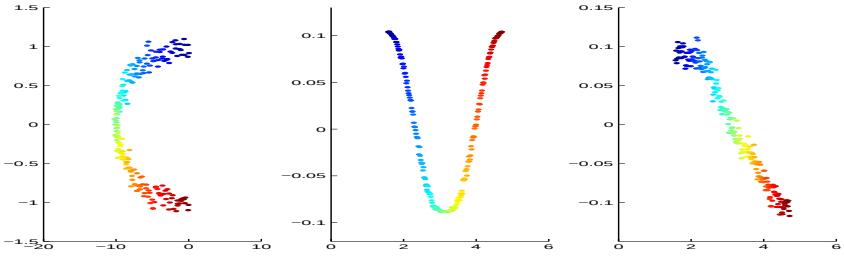

FIG. 11. *(left) Half ellipse data set, (middle) the global coordinates for the half ellipse data set, (right) the global coordinates after one iterative-refinement*

presented in [7].

2. A *discrete* version of the manifold learning can be formulated by considering the data points as the vertices of an undirected graph [6]. A quantization of the global coordinates specifies the adjacency relation of those vertices, and manifold learning becomes to discover whether an edge should be created between a pair of vertices or not so that the resulting vertex neighbors resemble those of the manifold. We need to investigate a proper formulation of the problem and the related optimization methods.

3. From a statistical point of view, it is also of great interest to investigate more precise formulation of the error model and the associated consistency issues and convergence rate as the sample size goes to infinity. The learn-ability of the nonlinear manifold also depends on the sampling density of the data points. Some of the results in non-parametric regression and statistical learning theory will be helpful to pursue this line of research.

REFERENCES

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimension reduction and data representation. Technical Report, Dept. of Statistics, Univ. of Chicago, 2001.
[2] D. Freedman. Efficient simplicial reconstructions of manifolds from their samples. *IEEE PAMI*, to appear, 2002.
[3] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 3nd edition, 1996.
[4] T. Hastie and W. Stuetzle. Principal curves. *J. Am. Statistical Assoc.*, 84: 502–516, 1988.
[5] T. Kohonen. *Self-organizing Maps*. Springer-Verlag, 3rd Edition, 2000.
[6] T. Martinetz and K. Schulten. Topology representing networks. *Neural Networks*, 7: 507–523, 1994.
[7] M. Polito and P. Perona. Grouping and Dimensionality reduction by Locally Linear Embedding. *Advances in Neural Information Processing Systems* 14, eds. T. Dietterich, S. Becker, Z.

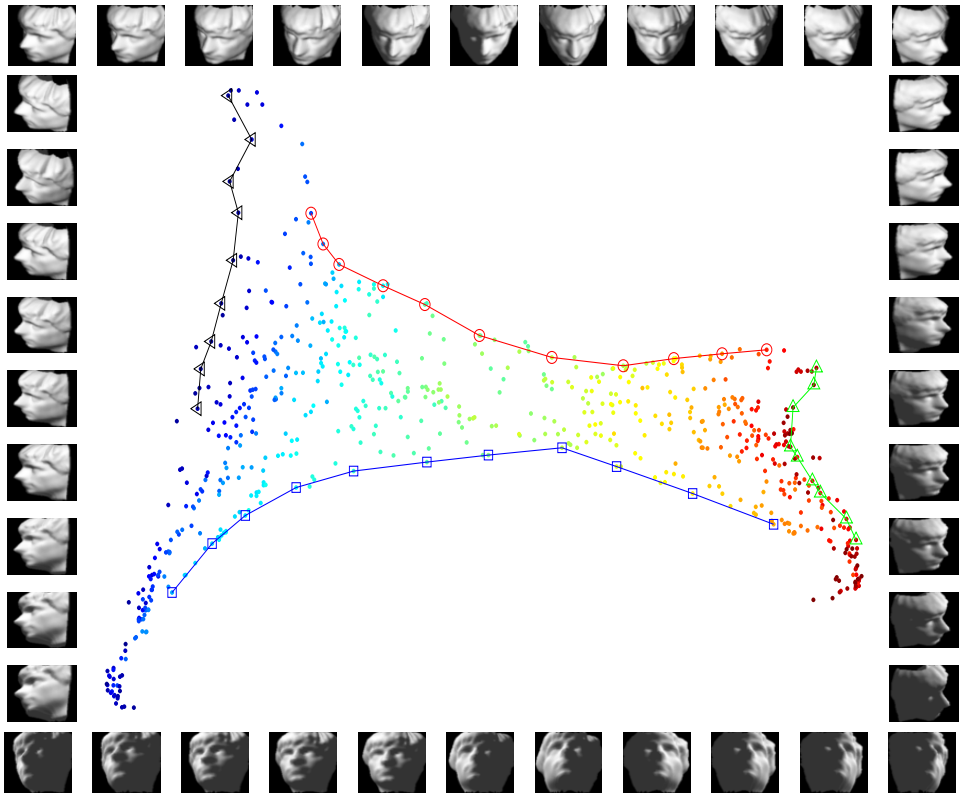

FIG. 12. *Coordinates computed by Algorithm LTSA with $k = 12$ neighbors (middle) and images corresponding to the points on the bound lines (top, bottom,left, and right) Left.*

Ghahramani, MIT Press (2002).
[8] S. Roweis and L. Saul. Nonlinear dimension reduction by locally linear embedding. *Science*, 290: 2323–2326, 2000.
[9] L. Saul and S. Roweis. Think globally, fit locally: unsupervised learning of nonlinear manifolds. Technical Reports, MS CIS-02-18, Univ. Pennsylvania, 2002.
[10] Y. Teh and S. Roweis. Automatic Alignment of Hidden Representations. To appear in *Advances in Neural Information Processing Systems*, 15, MIT Press (2003).
[11] J. Tenenbaum, V. De Silva and J. Langford. A global geometric framework for nonlinear dimension reduction. *Science*, 290:2319–2323, 2000.
[12] W.N. Venables and B.D. Ripley. Modern Applied Statistics with S-plus. Springer-verlag, 1999.

A. Alignment Preserving Orthogonality. Notice that the coordinates in the Θ_i's in (4.7) are obtained with respect to an orthonormal basis, therefore it seems quite natural to preserve this orthogonality in the low-dimensional feature space as well. To this end, we consider *orthogonal* transformation for alignment, i.e., we restrict the matrices L_i's in (4.7) to be orthogonal,

$$\tau_{i_j} = \bar{\tau}_i + G_i\theta_j^{(i)} + \epsilon_j^{(i)}, \quad G_i^T G_i = I_d, \tag{1.19}$$

and minimize the alignment error

$$\|E\|_F^2 \equiv \min_{T,\,\{G_i\}} \sum_i \|T_i(I - \frac{1}{k}ee^T) - G_i\Theta_i\|_F^2. \tag{1.20}$$

Note that here we do not impose the orthogonality constraints on T. Clearly for any fixed $T = T^*$, we have

$$\|E\|_F^2 \leq \min_{\{G_i\}} \sum_i \|T_i^*(I - \frac{1}{k}ee^T) - G_i\Theta_i\|_F^2,$$

giving an upper bound for the minimum. To obtain a tighter upper bound on $\|E\|_F^2$, we need the following lemma [3].

LEMMA A.1. *Let $AB^T = U\Sigma V^T$ be the SVD of AB^T with*

$$\Sigma = \operatorname{diag}(\sigma_1(AB^T), \cdots, \sigma_d(AB^T)).$$

Then the optimal orthogonal matrix G that minimizes $\|A - GB\|_F$ is given by $G = UV^T$. Furthermore,

$$\min_{G^TG=I} \|A - GB\|_F^2 = \|A\|_F^2 + \|B\|_F^2 - 2\sum_j \sigma_j(AB^T).$$

We now define $\hat{T}_i^* \equiv T_i^*(I - \frac{1}{k}ee^T)$. Similar to (6.16), we have

$$\hat{T}_i^* - G_i\Theta_i = \hat{T}_i^* - G_iP_i\hat{T}_i^* - Q_i^TE_i^*(I - \frac{1}{k}ee^T) + O(\|\hat{T}_i^*\|^2). \tag{1.21}$$

By LemmaA.1 and the inequalities $\sigma_i(AB) \geq \sigma_i(A)/\sigma_{\min}(B)$, we have

$$\begin{aligned}\min_{G_i^TG_i=I_d} \|\hat{T}_i^* - G_iP_i\hat{T}_i^*\|_F^2 &\leq \|\hat{T}_i^*\|_F^2 + \|P_i\hat{T}_i^*\|_F^2 - 2\sum_j \sigma_j(\hat{T}_i^*(P_i\hat{T}_i^*)^T) \\ &\leq \big(1 + \sigma_{\max}^2(P_i) - 2\sigma_{\min}^2(P_i)\big)\|\hat{T}_i^*\|_F^2.\end{aligned}$$

Therefore we obtain the following upper bound

$$\|E\|_F \leq \sum_i \big(1 + \sigma_{\max}^2(P_i) - 2\sigma_{\min}^2(P_i)\big)\|\hat{T}_i^*\|_F^2 + O\Big(\big(\sum_i \|\hat{T}_i^*\|_F^4\big)^{1/2}\Big).$$

The optimization problem (1.20) can be solved iteratively alternating between the following two steps.

1. For fixed $T_i, i = 1, \ldots, N$, minimize $\|T_i(I - \frac{1}{k}ee^T) - G_i\Theta_i\|$ to obtain an optimal G_i. By Lemma A.1, G_i is given by $G_i = U_iV_i^T$, where $T_i\Theta_i^T = U_i\Sigma_iV_i^T$.
2. For fixed $G_i, i = 1, \ldots, N$, solve the optimalization problem

$$\min_T \sum_i \|T_i(I - \frac{1}{k}ee^T) - G_i\Theta_i\|_F^2$$

to obtain a new T_i. This is a LS problem.

It converges monotonically.